

**PROPOSTA DE UM SISTEMA ESPECIALISTA  
EM PROJETOS ESTRUTURADOS**

**Ysmar Vianna e Silva Filho  
Antonio Anibal de Souza Teles  
Claudia Lage Rebello da Motta**

**NCE 0486**

**Novembro, 1986**

**Universidade Federal do Rio de Janeiro  
Núcleo de Computação Eletrônica  
Caixa Postal 2324  
20001 - Rio de Janeiro, RJ  
BRASIL**

**PROPOSTA DE UM SISTEMA ESPECIALISTA  
EM PROJETOS ESTRUTURADOS**

**Ysmar Vianna e Silva Filho  
Antonio Anibal de Souza Teles  
Claudia Lage Rebello da Motta**

**ABSTRACT**

This report suggests the development of an Expert System in Structured Design. Structured Techniques, the Expert System environment and the strategy of solution used are shown as well as final considerations about the Structured Design Techniques found in literature.

**RESUMO**

Este relatório apresenta a proposta de construção de um Sistema Especialista em Modularização de Projetos Estruturados. A Técnica de Projeto Estruturado, o ambiente de desenvolvimento escolhido e a estratégia de solução adotada são descritos, assim como considerações finais sobre o ambiente e sobre a literatura disponível relativa a Projetos Estruturados.

## 1. INTRODUÇÃO

As Técnicas de Inteligência Artificial são comumente empregadas na resolução de problemas que não possuam solução determinística com complexidade polinomial. Encontra-se na literatura referências a aplicações dessas técnicas nas áreas de Visão, Compreensão de Linguagem natural, Robótica e Jogos, entre outras.

Entretanto, a área de aplicação que vem popularizando essas técnicas é a de construção de Sistemas Especialistas, onde o conhecimento de um determinado domínio, geralmente baseado em probabilidades e heurísticas, é introduzido na Base de Conhecimentos do sistema.

O presente trabalho vem propor a construção de um Sistema Especialista em Modularização de Projetos Estruturados. Nele são apresentados o problemas em si, o ambiente utilizado para o desenvolvimento do sistema, a estratégia de solução adotada e algumas considerações sobre o ambiente e sobre as heurísticas relativas ao problema.

## 2. PROBLEMA - MODULARIZAÇÃO DE PROJETOS ESTRUTURADOS

O Projeto Estruturado é um conjunto de técnicas e de considerações que tornam mais simples, rápidas e portanto mais baratas as fases de projeto, codificação e depuração de programas [4,5,6]. Uma vez que a complexidade dos programas é reduzida, fica mais fácil a correção de erros, a modificação e a ampliação de funções já existentes, bem como o acréscimo de novas funções. E, sobretudo, possibilita a reutilização de códigos, característica mais útil a longo prazo.

Uma Especificação Estruturada é composta basicamente de dois elementos: o Diagrama de Fluxo de Dados (DFD) e o Dicionário de Dados (DD). Um DFD é um grafo direcionado que a

cada nó tem associado um processo de transformação de dados e que a cada aresta tem associado um conjunto de dados. O DD é empregado para documentar os dados do programa, sendo lá mantidos seus nomes, seus tipos, seus campos e os processos que utilizam.

Existem heurísticas que permitem dividir um DFD em três partes: uma associada à entrada e à adição de dados, outra associada à lógica do programa propriamente dita e a última associada à formatação e à preparação dos resultados. A partir desta divisão, pode-se construir o Diagrama Hierárquico de Módulos (DHM). O termo Módulo é usado para referenciar um conjunto de um ou mais comandos contíguos, possuindo um nome que pode ser utilizado em outras partes do programa onde seja necessária a sua ativação.

Como primeira aproximação, pode-se construir um DHM onde cada módulo corresponda a um nó do DFD. Um módulo A se comunica diretamente com um outro módulo B (A é pai de B ou B é pai de A) através da interface AB. Em qualquer interface existem dados e controles que são transferidos do módulo pai para o módulo filho assim como dados e controles que são transferidos no sentido inverso. Desta forma, um DHM é um grafo não direcionado onde cada nó corresponde a um módulo e cada aresta a uma interface entre dois módulos.

Existem heurísticas que visam a melhoria de um DHM. Estas heurísticas tratam principalmente das características dos módulos isoladamente, do relacionamento entre módulos, da natureza dos dados e da estrutura de um DHM como um todo.

Analisa-se um módulo medindo sua coesão, ou seja, verificando-se quão funcional o módulo é. Módulos pouco funcionais devem ser fatorados, ou agrupados, na tentativa de se criar módulos mais coesos.

Um aspecto ligado às interfaces entre módulos é verificar se um dado produzido por um módulo é consumido em módu-

los próximos a eles. Se isto não ocorrer, existirá casos em que um dado irá percorrer grande parte da estrutura, e portanto do programa, para atingir um módulo em que ele seja necessário. Outros problemas comuns são:

- a inversão de autoridade, onde um módulo inferior passa controles que comandam a execução de um ascendente;
- a existência de módulos funis, onde dados passam

O Sistema que está sendo proposto neste relatório tem por finalidade analisar a estruturação de um Projeto Estruturado. Recebendo como definição do projeto seu Diagrama Hierárquico de Módulos e seu Dicionário de Dados ele irá verificar heurísticamente se são atendidas as características de estruturação de um bom projeto, apresentando diagnósticos e sugerindo mudanças.

Pretende-se, desta forma, construir uma ferramenta inteligente que funcione como um consultor que auxilie Analistas de Sistemas que não possuam grande experiência em Projeto Estruturado.

### 3. AMBIENTE DE DESENVOLVIMENTO

Para o desenvolvimento do projeto será usado um Ambiente de Desenvolvimento de Sistemas Especialistas conhecido como ESE/VM. Um Ambiente de Desenvolvimento é um software que pode ser utilizado por especialistas, de diversas áreas do conhecimento, que possuam pequena experiência com o processamento de dados, para construir um Sistema Especialista. Um Ambiente deve aceitar informações descritivas, prover um mecanismo de inferência automático e ser capaz de justificar seu raciocínio durante uma consulta.

O ESE/VM (Expert System Environment) da IBM [1] é um sistema de propósito geral para a construção e execução de aplicações de Sistemas Especialistas. O ESE é composto de dois programas: um para o desenvolvimento de aplicações (ESDE - Expert System Development Environment) e o outro para a execução de tais aplicações (ESCE - Expert System Consultation Environment).

Para desenvolver um Sistema usando o ESE o especialista deve inserir uma base de conhecimento e escolher uma estratégia de inferência. A base de conhecimento é inserida através de um editor inteligente que verifica a consistência da informação editada, acusando erros sintáticos ou semânticos durante a própria edição. A escolha da estratégia de inferência irá basicamente determinar a ordem de aplicação das regras da base de conhecimentos. Duas estratégias são oferecidas: uma em que a obtenção de um valor para um objetivo é conseguida através do encadeamento para trás das regras até que as premissas sejam todas conhecidas e a outra em que considera-se somente as regras em cujas premissas só sejam referenciados valores conhecidos, fazendo-se um encadeamento para frente e descobrindo-se assim novos valores.

A base de conhecimento de uma aplicação do ESE contém informações na forma dos seguintes objetos:

- Parâmetros:
  - „ fatos da aplicação e suas restrições;
- Regras:
  - „ combinações de fatos que implicam em outros fatos;
- Blocos de Controle:
  - „ agrupamentos de regras e parâmetros que impõe uma organização da base de conhecimentos e especificações de controle;

- Grupos:

- . conjunto de regras, parâmetros ou blocos de controle;

- Telas:

- . telas padronizadas pelo construtor do Sistema Especialista para interação com o usuário.

Um parâmetro é um objeto que tem um nome, alguns atributos e que irá possuir um ou mais valores. Os principais atributos de um parâmetro são:

- seu tipo, que pode ser numérico, lógico, cadeia alfanumérica, cadeia binária ou cadeia hexadecimal; além disso, um parâmetro pode guardar um único valor ou muitos valores e também pode ser vetorado, sem limite explícito;
- uma sequência que determinará a forma que o valor do parâmetro será determinado; a obtenção do valor pode ser através de aplicação de regras, de pergunta ao usuário ou de execução de rotina externa provida pelo construtor do sistema;
- restrições impostas aos valores que o parâmetro pode receber.

As regras especificam relações entre parâmetros e são da forma IF (premissas) THEN (ações), fazendo com que ações sejam realizadas quando o resultado da avaliação das premissas seja verdadeiro. A cada parâmetro da base de conhecimento está associado um fator de confiabilidade que é utilizado nas regras e através destas pode ser propagado. As ações de uma regra são tipicamente atribuições, escritas em telas e alterações no controle da execução das regras.

Os blocos de controle oferecem um mecanismo eficiente para a modularização do sistema. Cada bloco contém um conjunto

de regras e parâmetros e representa uma unidade de trabalho a ser considerada durante a execução do sistema. Os blocos de controle são organizados hierarquicamente e cada um deles pode possuir múltiplas instâncias. Entre os principais atributos de um bloco de controle estão as listas de regras e de parâmetros que lhe pertencem, o número máximo de instâncias e seu texto, que contém as ações e as estratégias que serão realizadas quando da execução do bloco.

Os grupos são conjuntos de objetos similares, como parâmetros, regras e blocos de controle. A função de um grupo é prover um nome a um conjunto de objetos de modo a facilitar a referência a esse conjunto.

As telas servem para facilitar a interação do sistema com o usuário. Elas podem ser definidas durante a construção do sistema, o que irá proporcionar uma interação personalizada. A execução de uma escrita sem referência a uma tela particular produzirá uma saída em uma das telas padrões do sistema, de layout não muito interessante e com textos em inglês.

Rotinas externas, preferencialmente escritas em Pascal [2,3], podem ser agregadas ao ESE ampliando consideravelmente o poder de processamento de um Sistema Especialista. Desta forma é possível o acesso a arquivos, a Banco de Dados, a pacotes gráficos e ao CMS em geral.

Em Ambiente Personalizado de Desenvolvimento/Consulta pode ser criado escrevendo-se rotinas externas, que ampliem o ambiente original, e gerando-se um módulo executável que contenha o ESDE/ESCE e as rotinas da aplicação. Para que seja possível a troca de informações entre o ESDE/ESCE e as rotinas externas existe uma biblioteca de comunicação que permite às rotinas:

- conhecer o nome e o tipo do parâmetro ao qual será atribuído o valor retornado pela rotina;



- conhecer o número de argumentos passados à rotina;
- conhecer o nome e o tipo de cada argumento da rotina;
- conhecer o número de valores que um determinado argumento possua;
- acessar o valor do argumento (ou cada um de seus valores, se o argumento possuir mais de um) assim como o fator de confiabilidade de cada valor;
- enviar mensagens para o arquivo de "trace" da consulta;
- retornar um valor, ou uma série de valores, ao parâmetro objeto da execução da rotina, junto com seu(s) coeficiente(s) de confiabilidade.

#### 4. PROPOSTA DE SOLUÇÃO

A construção de um Sistema Especialista exige a captação dos conhecimentos de um determinado domínio e sua inserção na Base de Conhecimento do sistema. No caso particular do sistema de Análise de Modularização existe um volume razoável de textos abordando o assunto, o que a princípio, dispensa a necessidade de um especialista.

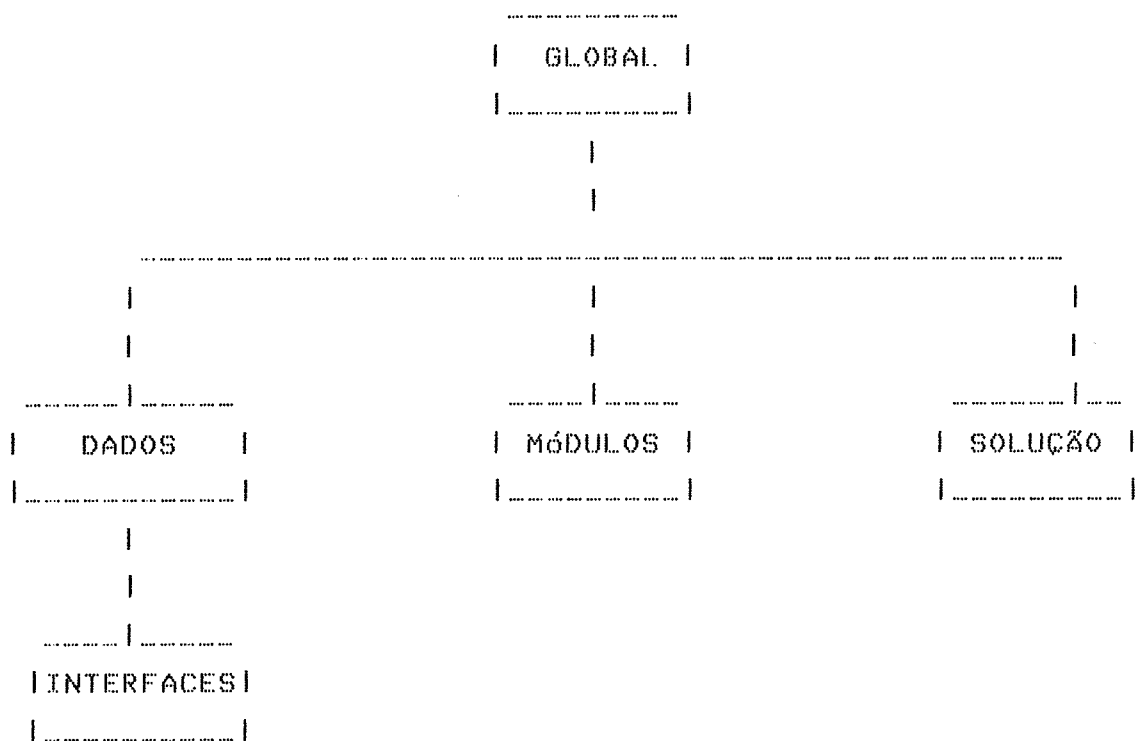
Estudando detalhadamente esses textos pode-se notar que o conhecimento é apresentado através das heurísticas que podem ser agrupadas em quatro categorias:

- a) Dados do sistema;
- b) Módulos que compõem o sistema;
- c) Interfaces entre os módulos;
- d) Estrutura geral do projeto.

A estratégia de solução proposta é dividir a análise em duas etapas. Na primeira, utiliza-se as heurísticas obtidas na literatura para fazer a coleta dos sintomas de possíveis problemas. Na segunda etapa faz-se um diagnóstico complexo e sugere-se um tratamento, que significará mudanças estruturais no projeto.

Esta estratégia é refletida na organização dos Blocos de Controle da implementação no ESE. Descrever a implementação do Sistema envolve a estruturação dos Blocos de Controle, a definição dos Parâmetros e a construção de Regras, bem como a alocação destes dois últimos nos Blocos.

Os Blocos de Controle são organizados hierarquicamente e essa hierarquia pode ser mostrada com o auxílio de uma árvore que possua um bloco Global como raiz.



Bloco GLOBAL - Pertencem a este bloco os parâmetros que guardam as características do projeto a ser criticado e os que guardam a série de sintomas detectados. Os parâmetros referentes ao projeto são preenchidos através da execução de ro-

tinhas externas do ESE, escritas em Pascal. Os sintomas são coletados durante a execução dos blocos Dados, Interfaces e Módulos. Após a execução desses blocos são considerados aspectos sobre o balanceamento da estrutura e é ativado o bloco Solução. Assim sendo, o texto de controle deste bloco é:

- a) Preencher os parâmetros com informações sobre o projeto;
- b) Executar o Bloco Dados;
- c) Executar o Bloco Módulos;
- d) Aplicar as regras referentes à estrutura.
- e) Executar o Bloco Solução.

Bloco DADOS - Na definição deste bloco diz-se que ele possui múltiplas instâncias, uma para cada dado do projeto que está sendo considerado. Pertencem a este bloco as regras referentes às heurísticas que tratam dos dados e, cada vez que esse bloco for instanciado, são preenchidos seus parâmetros que descrevem as características do dado em questão. Seu texto de controle é:

- a) Preencher os parâmetros referentes ao dado;
- b) Aplicar as regras, que fazem a coleta de sintomas;
- c) Executar o Bloco Interfaces (em que este dado flui);
- d) Executar nova instância do bloco Dados, se ainda houver algum dado a ser criticado.

Bloco INTERFACES - Este bloco também possui múltiplas instâncias, uma para cada interface entre dois módulos, por onde flui o dado que estiver sendo considerado pelo bloco DA-

DOS. Pertencem a este bloco as regras referentes às heurísticas que tratam do relacionamento entre módulos, e cada instância do bloco possuirá parâmetros com informações da interface em estudo. Seu texto de controle é:

- a) Preencher os parâmetros referentes à interface;
- b) Aplicar as regras que coletam sintomas;
- c) Executar nova instância do bloco Interfaces, se ainda houver alguma interface onde o dado em estudo flui.

Bloco MÓDULOS - Este bloco, como os anteriores, possui diversas instâncias, uma para cada módulos do projeto. Pertencem a este bloco as regras que implementam as heurísticas aplicáveis aos módulos, em particular as que analisam sua coesão. Os parâmetros do bloco, que descrevem um módulos, são preenchidos quando da ativação de uma instância. O texto de controle do bloco Módulos é:

- a) Preencher os parâmetros referentes ao módulo;
- b) Aplicar as regras do bloco;
- c) Executar nova instância do bloco Módulos, se ainda existir algum módulos a ser considerado.

Bloco SOLUÇÃO - Neste bloco são analisados os sintomas encontrados pelo sistema, muitos deles relacionados, e é feito um diagnóstico dos problemas que causaram esses sintomas. Uma vez localizado um problema, é sugerida uma mudança na forma de estruturação do projeto.

## 5. REPRESENTAÇÃO DO DHM

Um projeto estruturado é apresentado na forma de Diagrama Hierárquico de Módulos. Os dados manipulados pelo projeto estão no seu Dicionário de Dados. Um problema que se apresenta é como coletar e como armazenar esse volume significativo e razoavelmente complexo de informação nas estruturas de dados do ambiente que se pretende utilizar.

Como foi visto anteriormente, o ESE só permite que se defina parâmetros simples e vetorados, mesmo assim possuindo uma única dimensão. A solução encontrada foi utilizada inúmeros parâmetros vetorados (ORDERED) e representar as informações contendo um número variável de componentes através de pontos e de apontadores de início e fim desses componentes no ponto.

Tomando-se como exemplo a representação dos dados do projeto, teremos oito parâmetros vetorados, onde os seis primeiros estão relacionados entre si através de seus índices.

Nome ..... cadeia de caracteres  
Tipo ..... (DADO, CONTROLE, DESCRITIVO, HÍBRIDO)  
Complexidade ..... (SIMPLES, ESTRUTURA)  
Número-de-Campos ..... numérico  
Início-no-Poço ..... numérico  
Número-de-Elementos-no-Poço ..... numérico  
Poço-de-Interfaces ..... numérico  
Poço-de-Sentidos ..... (UP, DOWN, UD)

Desta forma, o dado de índice 1 pode ser descrito como:

Nome ..... ENDEREÇO  
Tipo ..... DADO  
Complexidade ..... ESTRUTURA  
Número-de-Campos ..... 7  
Início-no-Poço ..... 1  
Número-de-Elementos-no-Poço ..... 4

Os poços de Interfaces e de Sentidos teriam uma aparência semelhante a:

```
-----  
Poço-de-Interfaces = | 3 | 4 | 9 | 11 | ...  
                     |_____|_____|_____|_____|_____|
```

```
-----  
Poço-de-Sentidos   = | UP | UD | DOWN | DOWN | ...  
                     |_____|_____|_____|_____|_____|
```

Na representação anterior temos que o dado ENDEREÇO é uma estrutura com 7 campos (rua, número, complemento, bairro, cidade, estado, cep) e flui em quatro interfaces: Na interface 3 é passado do módulo filho para o módulo pai. Na 4 é passado do pai para o filho e devolvido (alterado) ao pai. Nas interfaces 9 e 11 é passado aos filhos sem ser devolvido ao pai (consumido pelos filhos ou por algum descendente destes).

Devido à grande quantidade de informações necessárias para a descrição de um DHM, optamos por descrever o projeto através de uma linguagem criada para este propósito. O texto é compilado e são gerados arquivos com formatos que simplificam o processo de carga dos parâmetros do ESE. A carga é feita através de rotinas escritas em Pascal, que são agregadas ao ambiente.

## 6. CODIFICAÇÃO DAS HEURÍSTICAS

O processo de aquisição do conhecimento foi dividido em etapas. Na primeira, as heurísticas encontradas na literatura são traduzidas para o português e escritas sem qualquer preocupação adicional. Como foram consultados vários textos, algumas heurísticas apareceram mais de uma vez, diferindo ape-

nas na forma. Na segunda etapa, são identificados os objetos que serão parâmetros nas regras. Cada parâmetro identificado recebe um nome e tem seu tipo definido. Na terceira etapa, as heurísticas são codificadas no formato das regras do ESE.

A esta altura do projeto a estrutura de Blocos de Controle já está concebida. Demos às regras, nomes que sugerem a que bloco cada uma delas irá pertencer. A alocação das regras nos blocos será feita de maneira indireta. Se a regra não for alocada explicitamente a um bloco, o ESE irá colocá-la no bloco de posição hierárquica mais inferior tal que todos os parâmetros referenciados na regra pertençam ao bloco ou a algum bloco hierarquicamente situado entre este e o bloco raiz da estrutura de controle. Terminada essa alocação automática, restará conferir se ela está feita da maneira prevista, e caso contrário fazer as adaptações necessárias.

Para exemplificar este processo, tomemos como exemplo a codificação a seguir.

Heurística: Um dado de controle que não é utilizado na lógica do módulo ao qual é passado como parâmetro sugere a existência de um "flag" que caminha pela estrutura do projeto.

PARÂMETRO	PERTENCE AO BLOCO
-----	-----
parm-tipo	Dados
parm-sentido	Interfaces
usado-lógica-filho	Interfaces
usado-lógica-pai	Interfaces
tramp-flag	Dados

Regra para o ESE:

```
if parm-tipo      = 'controle' and  
  ((parm-sentido = 'down' and usado-lógica-filho=false) or  
   (parm-sentido = 'up' and usado-lógica-pai=false))
```

then

There is 0.11 evidence that tramp-flag is true.

Nome da regra: INTER07 que será alocada no bloco INTERFACES.

## 7. INTERAÇÃO COM O USUÁRIO

O Sistema Especialista irá interagir com o usuário de três formas distintas. A modularização do projeto (DHM) é codificada na forma de um texto, que é compilado e posteriormente lido pelo Sistema. Durante a etapa de coleta de sintomas, alguns parâmetros do ESE são definidos como tendo seus valores determinados através de perguntas ao usuário. E finalmente, os diagnósticos e os tratamentos sugeridos são mostrados ao usuário empregando-se telas personalizadas. Como as duas últimas formas de interação ocorrem em situações que dispensam comentários, vamos passar à sintaxe da linguagem de descrição do projeto a ser criticado, o que é feito a seguir com o auxílio de diagramas sintáticos.



# PROJETO ::=

```

----> projeto ----> id ----> ; ----> L-MÓDULOS ----> DEF-INFO -----
                                     ↑           |   |
                                     |_____ |   |
                                     |_____ |
                                     |
|
|----> DEF-MOD ----> fim ----|
    ↑           |
    |_____ |

```

# L-MÓDULOS ::=

```

----> módulos ----> id ----> ; ----|
    ↑           |
    |--- , <-|

```

# DEF-INFO ::=

```

----> informação ----> id ----> # ----> { ----
|_____ |
|
|----> tipo ----> = ----> TIPO-ID ----> ; ----
|_____ |
|
|----> complexidade ----> = ----> COMPLEX-ID ----> ; ----
|_____ |
|
|----> número ----> de ----> campos ----> = ----> inteiro --
|_____ |
|_____ |
|_____ ; <-----|
|
|----> } ----|

```

# TIPO-ID ::=

```

----->      dado      <-----|
|               ↑
|----> controle  --->|
|               |
|----> descritivo --->|
|               |
|---->  híbrido   --->|

```

# COMPLEX-ID ::=

```

----->      simples  -----|
|               ↑
|----> estruturado ---|

```

# DEF-MOD ::=

```

----> módulo ----> id ----> # ----> ( ----
|
|
|----> pais ----> # ----> id ----> # ----> DEF-INTERF -----
|               ↑                                     | | |
|               |-----, <-----| |
|               |                                     |
|-----; <-----|
|
|----> filhos ----> # ----> id ----> # ----> DEF-INTERF -----
|               ↑                                     | | |
|               |-----, <-----| |
|               |                                     |
|-----; <-----|
|
|----> parâmetros ----> usados --> na --> lógica --> = --
|               |                                     |
|-----; <---- L-ID <-----|
|
|----> ) ----|

```

**DEF-INTERF ::=**

```

----> ( ----> ( ----> id ----> , ----> SENTIDO ----> ) ----> ) ----|
      ↑                                     |
      |----- , <-----|

```

**L-ID ::=**

```

-----> id -----|
      |               ↑
      |----- , <----|

```

**SENTIDO ::=**

```

-----> up -----|
      |               ↑
      |----> down ---->|
      |               |
      |----> ud ---->|

```

## 8. CONSIDERAÇÕES FINAIS

O sistema proposto neste relatório é fruto de um trabalho de oito meses. Durante este período foram realizadas as seguintes etapas:

- 1) Estudo das heurísticas sobre projeto estruturado encontradas na literatura;
- 2) Estudo do ambiente de desenvolvimento (ESE), seguido de inúmeras experiências;

- 3) Adoção da estratégia de solução anteriormente mostrada;
- 4) Definição e codificação dos objetos (Parâmetros, Regras e Textos de Controle) referentes aos blocos Global, Dados, Interfaces, Módulos e Solução;
- 5) Codificação e depuração das rotinas externas escritas em Pascal.

Durante esta fase foi possível fazer um análise crítica de dois fatores essenciais ao sistema: as heurísticas sobre projeto estruturado e o ambiente de desenvolvimento.

A respeito do ambiente (ESE) podemos citar suas principais virtudes e suas principais deficiências. Entre os aspectos positivos estão o fato do ambiente ser completo (possuindo facilidades de edição, verificação da consistência, execução de consultas e "trace"), de possuir uma estrutura de controle poderosa e de manipular informações incertas através de fatores de confiabilidade. Como principais deficiências estão a dificuldade de se representar informações complexas (como, por exemplo, grafos) e sua documentação, por demais descritiva e com pequena quantidade de exemplos. De um modo geral trata-se de uma ferramenta muito útil ao desenvolvimento de Sistemas Especialistas.

Os livros sobre projetos estruturados apresentam uma série de heurísticas para análise de modularização. O problema é que elas são muito vagas, pois estão baseadas em situações difíceis de serem determinadas. Ao estimar a coesão de um módulo, por exemplo, as heurísticas referem-se à existência de mais de uma função no módulo para determinar que ele não é funcional. O problema de aplicar essas heurísticas se reduz ao problema de determinar a existência de mais de uma função no módulo, o que em si é um outro problema grande de atacar. Além disso, não existe na literatura uma abordagem sistemática para a captura de conhecimentos de um especialista, o que dificulta

bastante a escrita das regras.

As próximas etapas de desenvolvimento do sistema incluirão o detalhamento das heurísticas, a depuração e avaliação da base de conhecimentos e a elaboração de casos de teste. Desta forma pretende-se ampliar consideravelmente a "inteligência" do sistema.

## 9. AGRADECIMENTOS

A utilização do ambiente ESE foi possível graças ao apoio da IBM BRASIL que nos permitiu acesso aos equipamentos e às dependências do seu Centro Educacional e Residencial da Gávea. Em particular, nossos agradecimentos a Carlos Roberto de Souza, pesquisador do Centro Científico da IBM, pelo auxílio e paciência que nos dedicou.

## REFERÊNCIAS:

### 01. IBM

Expert System Development Environment/VM  
Reference Manual

### 02. IBM

Pascal/VS - Language Reference Manual  
1985;

### 03. IBM

Pascal/VS - Programmer's Guide  
1985;

### 04. MEILIR PAGE - JONES

The Practical Guide to Struct. Systems Design  
Yourdon - Press, 1980;

05. WAYNE P. STEVENS

Projeto Estruturado de Sistemas

Editora Campos, 1985;

06. YOURDON, EDWARD & LARRY L. CONSTANTINE

Structured Design

Prentice - Hall, 1979.